1 1. (original) A method of providing concurrent access to a resource object, the method

2 comprising the computer-implemented steps of:

3 creating and storing a lock data structure for a particular resource object, the lock data

4 structure comprising data indicative of values for a resource object

5 identification, a lock type, and a version number related to a number of

6 changes to the resource object since the lock data structure was generated;

7 receiving a request from a requesting process for a requested lock type for access to

8 the particular resource object; and

9 determining whether to grant the request based on the requested lock type and the

10 lock type in the lock data structure.

1 2. (original) A method as recited in Claim 1, further comprising the step of:

2 if it is determined to grant the request, then

3 bringing the value of the lock type in the data structure into agreement with

4 the lock type in the request;

5 generating a lock object having data indicative of the values for the resource

6 object identification, the lock type and the version number from the

7 lock data structure, and

8 returning the lock to the requesting process.

1 3. (original) A method as recited in Claim 1, further comprising the steps of:

2 receiving a lock to be released having data indicative of values for the resource object

3 identification and the lock type and the version number;

2

4		determining whether the data indicative of the value for the lock type in the lock to be
5		released indicates an exclusive lock, and
6		if it is determined the data indicates the exclusive lock is to be released, then changing
7		the value for the version number in the lock data structure based on the value
8		of the version number in the lock to be released.

1	4.	(original) A method as recited in Claim 2, wherein:
2		the lock data structure further comprises a reference number;
3		said step of generating a lock data structure further comprises setting the reference
4		number to a predetermined initial value; and
5		said method further comprises, if it is determined to grant the request, then replacing
6		the value of the reference number in the lock data structure with a sum of the
7		value of the reference number in the lock data structure and a predetermined
8		reference change value.

1	5.	(original) A method as recited in Claim 4, further comprising the steps of:
2		receiving a lock to be released having data indicating the particular resource object;
3		determining whether the reference number substantially equals the predetermined
4		initial value of the reference number; and
5		if it is determined the reference number does not substantially equal the
6		predetermined initial value, then replacing the value of the reference number
7		in the lock data structure with a difference substantially equal to the value of
8		the reference number in the lock data structure minus the predetermined
9		reference change.

1   6.   (original) A method as recited in Claim 5 further comprising, if it is determined the

2   reference substantially equals the predetermined initial value, then deleting the lock data

3   structure for the particular resource object.

1   7.   (canceled)

1   8.   (currently amended) <u>A method of updating a resource object using optimistic locks,</u>

2   <u>the method comprising the computer-implemented steps of:</u> ~~The method as recited in Claim~~

3   ~~7, said using the optimistic lock comprising:~~

4   <u>receiving from a client process a request to update a particular resource object;</u>

5   <u>sending to a lock manager process a request for a first lock for access to the particular</u>

6   <u>resource object, the request including data indicating an optimistic lock type;</u>

7   <u>receiving the first lock for access to the particular resource object, the first lock</u>

8   <u>including data indicating the resource object, the optimistic lock type and a</u>

9   <u>first value for a version number related to a number of changes to the resource</u>

10   <u>object since the lock manager generated a lock data structure corresponding to</u>

11   <u>the resource object; and</u>

12   <u>updating the resource object by</u>

13   <u>sending to a lock manager process a request for a second lock for access to the</u>

14   <u>particular resource object, the request including data indicating the</u>

15   <u>resource object identification and an exclusive lock type;</u>

16   <u>receiving the second lock for access to the particular resource object, the</u>

17   <u>second lock including data indicating the resource object</u>

4

18   identification, the exclusive lock type and a second value for the

19   version number;

20   determining whether the second value for the version number substantially

21   equals the first value for the version number; and

22   if the second value substantially equals the first value, then

23   committing an updated resource object to the resource, and

24   replacing the second value in the reference number in the second lock

25   with a third value of the version number, the third value

26   computed by adding the second value and a predetermined

27   version change value.

1   9.   (original) The method as recited in Claim 8, further comprising, if the second value

2   does not substantially equal the first value, then sending a message to the client process, the

3   message indicating that the resource object was not updated.

1   10.   (currently amended) The method as recited in Claim ~~7~~ 8, further comprising sending

2   to the lock manager process a first release message to release the first lock.

1   11.   (currently amended) The method as recited in Claim ~~8~~ 10, further comprising sending

2   to the lock manager process a second release message to release the second lock.

1   12.   (currently amended) The method as recited in Claim 9, further comprising sending to

2   the lock manager process a ~~second~~ release message to release the second lock, the ~~second~~

3   release message including data indicating the third value of the version number in the second

4   lock and the exclusive lock type, wherein the third value of the version number is used by the

5   lock manager to replace the second value of the version number in the lock data structure.

1   13.   (original) A computer-readable medium carrying one or more sequences of

2   instructions for providing concurrent access to a resource object, which instructions, when

3   executed by one or more processors, cause the one or more processors to carry out the steps

4   of:

5   generating a lock data structure for a particular resource object, the lock data structure

6   comprising data indicative of values for a resource object identification, a lock

7   type, and a version number related to a number of changes to the resource

8   object since the lock data structure was generated;

9   receiving a request from a requesting process for a requested lock type for access to

10   the particular resource object; and

11   determining whether to grant the request based on the requested lock type and the

12   lock type in the lock data structure.

1   14.   (currently amended) A computer-readable medium carrying one or more sequences of

2   instructions for updating a resource object, which instructions, when executed by one or more

3   processors, cause the one or more processors to carry out the steps of:

4   receiving from a client process a request to update a particular resource object;

5   sending to a lock manager process a request for a first lock for access to the particular

6   resource object, the request including data indicating an optimistic lock type;

7   receiving the first lock for access to the particular resource object, the first lock

8   including data indicating the resource object, the optimistic lock type and a

9   first value for a version number related to a number of changes to the resource

10   object since the lock manager generated a lock data structure corresponding to

11   the resource object; and

6

12 ~~using the optimistic lock to update~~ updating the resource object by

13   sending to a lock manager process a request for a second lock for access to the

14   particular resource object, the request including data indicating the

15   resource object identification and an exclusive lock type;

16   receiving the second lock for access to the particular resource object, the

17   second lock including data indicating the resource object

18   identification, the exclusive lock type and a second value for the

19   version number;

20   determining whether the second value for the version number substantially

21   equals the first value for the version number; and

22   if the second value substantially equals the first value, then

23   committing an updated resource object to the resource, and

24   replacing the second value in the reference number in the second lock

25   with a third value of the version number, the third value

26   computed by adding the second value and a predetermined

27   version change value.

1   15.   (original) An apparatus for providing concurrent access to a resource object,

2 comprising:

3   a processor;

4   one or more stored sequences of instructions which, when executed by the processor,

5   cause the processor to carry out the steps of:

6   generating a lock data structure for a particular resource object, the lock data

7   structure comprising data indicative of values for a resource object

8   identification, a lock type, and a version number related to a number of

9   changes to the resource object since the lock data structure was

10   generated;

11   receiving a request from a requesting process for a requested lock type for

12   access to the particular resource object; and

13   determining whether to grant the request based on the requested lock type and

14   the lock type in the lock data structure.

1   16.   (currently amended) An apparatus for updating a resource object, comprising:

2   a processor;

3   one or more stored sequences of instructions which, when executed by the processor,

4   cause the processor to carry out the steps of:

5   receiving from a client process a request to update a particular resource object;

6   sending to a lock manager process a request for a first lock for access to the

7   particular resource object, the request including data indicating an

8   optimistic lock type;

9   receiving the first lock for access to the particular resource object, the first

10   lock including data indicating the resource object, the optimistic lock

11   type and a first value for a version number related to a number of

12   changes to the resource object since the lock manager generated a lock

13   data structure corresponding to the resource object; and

14   ~~using the optimistic lock to update~~ updating the resource object <u>by</u>

15   <u>sending to a lock manager process a request for a second lock for</u>

16   <u>access to the particular resource object, the request including</u>

8

17   data indicating the resource object identification and an

18   exclusive lock type;

19   receiving the second lock for access to the particular resource object,

20   the second lock including data indicating the resource object

21   identification, the exclusive lock type and a second value for

22   the version number;

23   determining whether the second value for the version number

24   substantially equals the first value for the version number; and

25   if the second value substantially equals the first value, then

26   committing an updated resource object to the resource, and

27   replacing the second value in the reference number in the

28   second lock with a third value of the version number,

29   the third value computed by adding the second value

30   and a predetermined version change value.

1   17.   (original) An apparatus for providing concurrent access to a resource object,

2   comprising:

3   means for generating a lock data structure for a particular resource object, the lock

4   data structure comprising data indicative of values for a resource object

5   identification, a lock type, and a version number related to a number of

6   changes to the resource object since the lock data structure was generated;

7   means for receiving a request from a requesting process for a requested lock type for

8   access to the particular resource object; and

9 means for determining whether to grant the request based on the requested lock type

10 and the lock type in the lock data structure.

1 18. (currently amended) An apparatus for updating a resource object, comprising:

2 a means for receiving from a client process a request to update a particular resource

3 object;

4 a means for sending to a lock manager process a request for a first lock for access to

5 the particular resource object, the request including data indicating an

6 optimistic lock type;

7 a means for receiving the first lock for access to the particular resource object, the

8 first lock including data indicating the resource object, the optimistic lock type

9 and a first value for a version number related to a number of changes to the

10 resource object since the lock manager generated a lock data structure

11 corresponding to the resource object; and

12 a means for ~~using the optimistic lock to update~~ updating the resource object,

13 including

14 a means for sending to a lock manager process a request for a second lock for

15   access to the particular resource object, the request including data

16   indicating the resource object identification and an exclusive lock type;

17 a means for receiving the second lock for access to the particular resource

18   object, the second lock including data indicating the resource object

19   identification, the exclusive lock type and a second value for the

20   version number;

21 <u>a means for determining whether the second value for the version number</u>

22   <u>substantially equals the first value for the version number;</u>

23 <u>a means for committing an updated resource object to the resource if the</u>

24   <u>second value substantially equals the first value; and</u>

25 <u>a means for replacing the second value in the reference number in the second</u>

26   <u>lock with a third value of the version number if the second value</u>

27   <u>substantially equals the first value, the third value computed by adding</u>

28   <u>the second value and a predetermined version change value.</u>

1 19. (new) The computer-readable medium as recited in Claim 14, wherein the

2 instructions, when executed by one or more processors, cause the one or more processors to

3 carry out the step of:

4   if the second value does not substantially equal the first value, then sending a message

5   to the client process, the message indicating that the resource object was not

6   updated.

1 20. (new) The computer-readable medium as recited in Claim 14, wherein the

2 instructions, when executed by one or more processors, cause the one or more processors to

3 carry out the step of:

4   sending to the lock manager process a first release message to release the first lock.

1 21. (new) The computer-readable medium as recited in Claim 20, wherein the

2 instructions, when executed by one or more processors, cause the one or more processors to

3 carry out the step of:

4   sending to the lock manager process a second release message to release the second

5   lock.

1   22.   (new) The computer-readable medium as recited in Claim 19, wherein the

2 instructions, when executed by one or more processors, cause the one or more processors to

3 carry out the step of:

4   sending to the lock manager process a release message to release the second lock, the

5   release message including data indicating the third value of the version

6   number in the second lock and the exclusive lock type, wherein the third value

7   of the version number is used by the lock manager to replace the second value

8   of the version number in the lock data structure.

1   23.   (new) The apparatus as recited in Claim 16, wherein the instructions, when executed

2 by one or more processors, cause the one or more processors to carry out the step of:

3   if the second value does not substantially equal the first value, then sending a message

4   to the client process, the message indicating that the resource object was not

5   updated.

1   24.   (new) The apparatus as recited in Claim 16, wherein the instructions, when executed

2 by one or more processors, cause the one or more processors to carry out the step of:

3   sending to the lock manager process a first release message to release the first lock.

1   25.   (new) The apparatus as recited in Claim 24, wherein the instructions, when executed

2 by one or more processors, cause the one or more processors to carry out the step of:

3   sending to the lock manager process a second release message to release the second

4   lock.

1   26.   (new) The apparatus as recited in Claim 23, wherein the instructions, when executed

2   by one or more processors, cause the one or more processors to carry out the step of:

3   sending to the lock manager process a release message to release the second lock, the

4   release message including data indicating the third value of the version

5   number in the second lock and the exclusive lock type, wherein the third value

6   of the version number is used by the lock manager to replace the second value

7   of the version number in the lock data structure.

1   27.   (new) The apparatus as recited in Claim 18, further comprising:

2   means for sending a message to the client process if the second value does not

3   substantially equal the first value, the message indicating that the resource

4   object was not updated.

1   28.   (new) The apparatus as recited in Claim 18, further comprising:

2   means for sending to the lock manager process a first release message to release the

3   first lock.

1   29.   (new) The apparatus as recited in Claim 28, further comprising:

2   means for sending to the lock manager process a second release message to release

3   the second lock.

1   30.   (new) The apparatus as recited in Claim 27, further comprising:

2   means for sending to the lock manager process a second release message to release the

3   second lock, the second release message including data indicating the third value

4   of the version number in the second lock and the exclusive lock type, wherein

the third value of the version number is used by the lock manager to replace the second value of the version number in the lock data structure.